HotSpot虚拟机对象探秘

2021年12月30日 11:57

Java对象创建过程.

1：检查类是否已经被加载；
2：为对象分配内存空间；

①内存分配方式.

②内存分配并发问题.

内存分配：对象引用指向内存区域.

解决方案：
- CAS+重试机制. → 是线程独享的，分配的时候不会冲突.
- TLAB：为每一个线程预先在 Eden 区分配一块儿内存，JVM 在给线程中的对象分配内存 时，首先在 TLAB 分配，当对象大于 TLAB 中的剩余内存或 TLAB 的内存已用尽时，再采用 上述的 CAS 进行内存分配

—XX：useTLAB.

3：为分配的内存空间初始化零值（为对象字段设置零值）；
4：对对象进行其他设置（设置对象头）；
5：执行构造方法。

对象的内存布局.

对象内存结构.
- 对象头.
- 实例数据.
- 对齐填充.

①对象头.

对象头里主要包括几类信息，分别是锁状态标志、持有锁的线程ID、，GC分代年龄、对象HashCode，类元信息地址、数组长度，这里并没有对对象头里的每个信息都列出而是进行大致的分类，下面是对其中几类信息进行说明。

锁状态标志： 对象的加锁状态分为无锁、偏向锁、轻量级锁、重量级锁几种标记。
持有锁的线程： 持有当前对象锁定的线程ID。
GC分代年龄： 对象每经过一次GC还存活下来了，GC年龄就加1。
类元信息地址： 可通过对象找到类元信息，用于定位对象类型。
数组长度： 当对象是数组类型的时候会记录数组的长度。

②实例数据.

对象实例数据才是对象的自身真正的数据，主要包括自身的成员变量信息，同时还包括实现的接口、父类的成员变量信息。

③对齐填充.

根据JVM规范对象申请的内存地址必须是8的倍数，换句话说对象在申请内存大小时候8字节的倍数，如果对象自身的信息大小没有达到申请的内存大小，那么这部分是对剩余部分进行填充。

对象的访问定位.

Java通过操作栈上的reference操作堆上的具体对象.

①定位方式

对象的访问定位.

Jam通过操作栈上的reference操作堆上的具体对象.

① 定位方式.

(1) 句柄. 对象移动时不会修改reference.

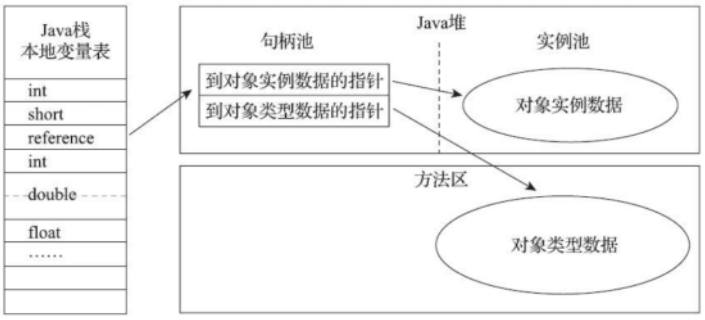

图2-2 通过句柄访问对象

(2) 直接指针访问. 少1次间接访问.

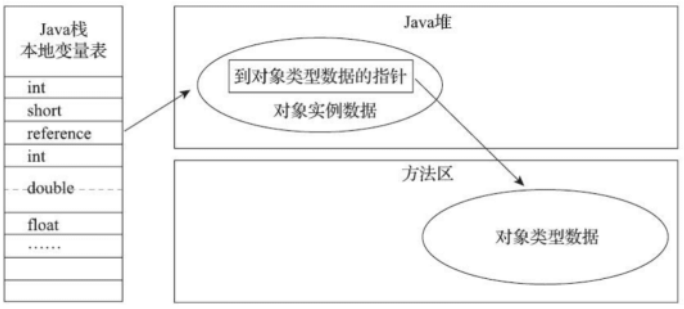

图2-3 通过直接指针访问对象

OOM异常.

-XX:+HeapDumpOnOutOfMemoryError: 打印内存异常日志.

① 排查.

(1) 分辨是 内存泄漏 or 内存溢出.

若是内存泄漏 查看 GC Roots 引用链.

若是内存溢出 看能否增大内存、能否内存优化.

② Stack Overflow.

-Xss 设置栈大小.

③ 方法区.

-XX:MaxMetaspaceSize.

-XX:MetaspaceSize.

-XX:MinMetaspaceFreeRatio.

④ 直接内存.

-XX:MaxDirectMemorySize.